

JAVA操作数据库方式与设计模式应用 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/144/2021_2022_JAVA_E6_93_8D_E4_BD_9C_c104_144427.htm 1. 在业务层使用JDBC直接操作数据库 - 最简单，最直接的操作

1) 数据库

url,username,password写死在代码

```
Class.forName("oracle.jdbc.driver.OracleDriver").newInstance().
```

```
String url="jdbc:oracle:thin:@localhost:1521:orcl". String
```

```
user="scott". String password="tiger". Connection conn=
```

```
DriverManager.getConnection(url,user,password). Statement
```

```
stmt=conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE
```

```
,ResultSet.CONCUR_UPDATABLE). String sql="0select * from
```

```
test". ResultSet rs=stmt.executeQuery(sql). 2) 采用Facade
```

和Command模式，使用DBUtil类封装JDBC操作；数据库

url,username,password可以放在配置文件中（

如xml,properties,ini等）。这种方法在小程序中应用较多

。2.DAO(Data Accessor Object)模式 - 松耦合的开始DAO =

data accessor domain object例如User类 - domain object

(javabean)UserDAO类 - accessor，提供的方法getUser(int id)

，save(User user)内包含了JDBC操作在业务逻辑中使用这两个

类来完成数据操作。使用Factory模式可以方便不同数据库连接

之间的移植。3.数据库资源管理模式3.1 数据库连接池技术

资源重用，避免频繁创建，释放连接引起大量性能开销；

更快的系统响应速度；通过实现JDBC的部分资源对象接口(

Connection, Statement, ResultSet)，可以使用Decorator设计模式

分别产生三种逻辑资源对象: PooledConnection,

PooledStatement和 PooledResultSet。 一个最简单地数据库连接池实现：

```
public class ConnectionPool {private static Vector
pools.private final int POOL_MAXSIZE = 25./** 获取数据库连接* 如果当前池中有可用连接，则将池中最后一个返回；若没有，则创建一个新的返回*/public synchronized Connection
getConnection() {Connection conn = null.if (pools == null) {pools
= new Vector().}if (pools.isEmpty()) {conn = createConnection().}
else {int last_idx = pools.size() - 1.conn = (Connection)
pools.get(last_idx).pools.remove(last_idx).}return conn.}/** 将使用完毕的数据库连接放回池中* 若池中连接已经超过阈值，则关闭该连接；否则放回池中下次再使用*/public synchronized
void releaseConnection(Connection conn) {if (pools.size() >=
POOL_MAXSIZE)try {conn.close().} catch (SQLException e) {//
TODO自动生成 catch 块e.printStackTrace().}
elsepools.add(conn).}public static Connection createConnection()
{Connection conn = null.try
{Class.forName("oracle.jdbc.driver.OracleDriver").newInstance().St
ring url = "jdbc:oracle:thin:@localhost:1521:orcl".String user =
"scott".String password = "tiger".conn =
DriverManager.getConnection(url, user, password).} catch
(InstantiationException e) {// TODO自动生成 catch
块e.printStackTrace().} catch (IllegalAccessException e) {// TODO
自动生成 catch 块e.printStackTrace().} catch
(ClassNotFoundException e) {// TODO自动生成 catch
块e.printStackTrace().} catch (SQLException e) {// TODO自动生
成 catch 块e.printStackTrace().}return conn.}}注意：利
```

用getConnection()方法得到的Connection，程序员很习惯地调用conn.close()方法关闭了数据库连接，那么上述的数据库连接机制便形同虚设。在调用conn.close()方法方法时如何调用releaseConnection()方法？这是关键。这里，我们使用Proxy模式和java反射机制。

```
public synchronized Connection
getConnection() {Connection conn = null.if (pools == null) {pools
= new Vector().}if (pools.isEmpty()) {conn = createConnection().}
else {int last_idx = pools.size() - 1.conn = (Connection)
pools.get(last_idx).pools.remove(last_idx).}ConnectionHandler
handler=new ConnectionHandler(this).return
handler.bind(con).}public class ConnectionHandler implements
InvocationHandler {private Connection conn.private
ConnectionPool pool.public ConnectionHandler(ConnectionPool
pool){this.pool=pool.}/** 将动态代理绑定到指定Connection*
@param conn* @return*/public Connection bind(Connection
conn){this.conn=conn.Connection
proxyConn=(Connection)Proxy.newProxyInstance(conn.getClass(
).getClassLoader(), conn.getClass().getInterfaces(),this).return
proxyConn.}/* (非 Javadoc) * @see
java.lang.reflect.InvocationHandler#invoke(java.lang.Object,
java.lang.reflect.Method, java.lang.Object[])*/public Object
invoke(Object proxy, Method method, Object[] args) throws
Throwable {// TODO自动生成方法存根Object
obj=null.if("close".equals(method.getName())){this.pool.releaseCo
nnection(this.conn).}else{obj=method.invoke(this.conn,
args).}return obj.}}
```

在实际项目中，并不需要你来从头开始来设

计数据库连接池机制，现在成熟的开源项目，如C3P0,dbcp,Proxool等提供了良好的实现。一般推荐使用Apache dbcp，基本使用实例：

```
DataSource ds = null;try{Context initCtx = new InitialContext();Context envCtx = (Context) initCtx.lookup("java:comp/env");ds = (DataSource)envCtx.lookup("jdbc/myoracle");if(ds!=null){out.println("Connection is OK!");Connection cn=ds.getConnection();if(cn!=null){out.println("cn is Ok!");Statement stmt = cn.createStatement();ResultSet rst = stmt.executeQuery("0select * from BOOK");out.println(" rst is Ok!");rst.next();while(rst.next()){out.println(" BOOK_CODE:" + rst.getString(1));}cn.close();}else{out.println("rst Fail!");}else out.println("Fail!");}catch(Exception ne){ out.println(ne);}3.2 Statement Pool普通预编译代码：String strSQL= " 0select name from items where id=? " .PreparedStatement ps=conn.prepareStatement(strSQL).ps.setString(1, " 2 " ).ResultSet rs=ps.executeQuery().但是PreparedStatement 是与特定的Connection关联的，一旦Connection关闭，则相关的PreparedStatement 也会关闭。为了创建PreparedStatement 缓冲池，可以在invoke方法中通过sql语句判断池中还有没有可用实例。

#### 4. 持久层设计与O/R mapping 技术



1) HERNATE：适合对新产品的开发，进行封闭化的设计 Hibernate 2003年被Jboss接管，通过把java pojo对象映射到数据库的table中，采用了xml/javareflection技术等。3.0提供了对存储过程和手写sql的支持，本身提供了hql语言。开发所需要的文件：hibernate配置文件：hibernate.cfg.xml 或 hibernate.propertieshibernate 映射


```

文件：a.hbm.xml pojo类源文件：a.java 导出表与表之间的关系：
a. 从java对象到hbm文件：xdoclet b. 从hbm文件到java对象：
hibernate extension c. 从数据库到hbm文件：middlegen d.
从hbm文件到数据库：SchemaExport2) Iatis：适合对遗留系统的改造和对既有数据库的复用，有很强的灵活性 3)
Apache OJB：优势在于对标准的全面支持 4) EJB：适合集群服务器，其性能也不象某些人所诟病的那么差劲 5) JDO
(java data object) 设置一个Properties对象，从而获取一个JDO的PersistenceManagerFactory（相当于JDBC连接池中的DataSource），进而获得一个PersistenceManager对象（相当于JDBC中的Connection对象），之后，你可以用这个PersistenceManager对象来增加、更新、删除、查询对象。
JDOQL是JDO的查询语言；它有点象SQL，但却是依照Java的语法的。
5. 基于开源框架的Struts Spring Hibernate实现方案
示例：这是一个3层架构的web程序，通过一个Action来调用业务代理，再通过它来回调DAO类。下面的流程图表示了MyUsers是如何工作的。数字表明了流程的先后顺序，从web层(UserAction)到中间层(UserManager)，再到数据层(UserDAO)，然后返回。Spring是AOP, UserManager和UserDAO都是接口。
1) web层(UserAction)：调用中间层的接口方法，将UserManager作为属性注入。采用流行的Struts框架，虽然有很多人不屑一顾，但是这项技术在业界用的比较普遍，能满足基本的功能，可以减少培训学习成本。
2) 中间层(UserManager)：将UserDAO作为属性注入，其实现主要是调用数据层接口的一些方法；它处于事务控制中。采用Spring框架实现，IOC与AOP是它的代名词，功能齐全，非常棒的

一个架构。3) 数据层(UserDAO)：实现类继承HibernateDaoSupport类，在该类中可以调用getHibernateTemplate()的一些方法执行具体的数据操作。采用Hibernate做O/R mapping，从种种迹象可以看出，Hibernate就是EJB3.0的beta版。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com